IDA

INSTITUTE FOR DEFENSE ANALYSES

Applications of Stochastic Analyses for Collaborative Learning and Cognitive Assessment

Amy Soller
Ron Stevens – UCLA

April 2007

IDA Document D-3421
Log: H 07-000947

This work was conducted under IDA's central research program, CRP 2112. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

INSTITUTE FOR DEFENSE ANALYSES

IDA Document D-3421

Applications of Stochastic Analyses for Collaborative Learning and Cognitive Assessment

Amy Soller
Ron Stevens – UCLA

PREFACE

This work was supported by a Central Research Program (CRP) task at the Institute for Defense Analyses (IDA).

This document will be published as a chapter in *Advances in Latent Variable Mixture Models,* Gregory R. Hancock and Karen M. Samuelsen (Eds.), Information Age Publishing, forthcoming 2007.

ACKNOWLEDGMENTS

The authors thank Dr. Greg Hancock, Dr. Karen Samuelson, Dr. Shelley Cazares, Dr. Mike Rigdon, and Mr. John Everett for their invaluable help in preparing and reviewing this chapter.

CONTENTS

FIGURES

EXECUTIVE SUMMARY

This paper presents a basic introduction to some popular stochastic analysis methods from an unbiased disciplinary perspective. Examples ranging from fields as diverse as defense analysis, cognitive science, and instruction are illustrated throughout to demonstrate the variety of applications that benefit from such stochastic analysis methods and models. Two applications of longitudinal stochastic analysis methods to collaborative and cognitive training environments are discussed in detail. The first application applies a combination of latent mixed Markov modeling and multidimensional scaling for modeling, analyzing, and supporting the process of online knowledge sharing. In the second application, a combination of iterative nonlinear machine learning algorithms is applied to identify latent classes of problem-solving strategies.

The examples illustrated in this paper are instances of an increasing global trend toward interdisciplinary research. As this trend continues to grow, research that takes advantage of the gaps and overlaps in analytical methodologies between disciplines will save time, effort, and research funds.

Applications of Stochastic Analyses for Collaborative Learning and Cognitive Assessment

The growing trend in interdisciplinary graduate programs appropriately breeds small developing communities of researchers who close the gaps between mainstream fields such as artificial intelligence, statistics, engineering, cognitive science, and psychology. The result is twofold and realizes both advantages and disadvantages. While we may discover new applications of existing algorithms and methods, we may also uncover similarities among the algorithms and methods that cross discipline boundaries and realize that some effort has been wasted in reinventing the wheel. As an alumnus of an interdisciplinary graduate program in applied artificial intelligence, the first author notes that she is often struck by the distance between communities that concurrently develop and apply identical methods to solve entirely different problems.

In this paper, we explore statistical and artificial intelligence perspectives on the field of stochastic sequence analysis. This field is particularly susceptible to the cross-discipline effect because of the wide array of analytical possibilities for application. Examples throughout this paper, from fields as diverse as defense analysis, cognitive science, and instruction, demonstrate the variety of applications that benefit from such stochastic analysis methods and models.

We begin with an introduction to probabilistic sequential class analysis aimed at addressing the terminological inconsistencies among the applied artificial intelligence, statistics, and educational psychology communities. This introduction helps explain why, for example, the literature in the applied artificial intelligence and biology communities on hidden Markov models (HMMs) is largely separate from the comparable body of literature in the sociological and psychological measurement and statistics communities on latent Markov models (Visser, Maartje, Raijmakers, & Molenaar, 2002).

The second part of this paper illustrates two applications of the methods described in the first part. The first application, Encouraging Positive Social Interaction while Learning ON-Line (EPSILON), employs a combination of latent mixed Markov modeling and multidimensional scaling (MDS) for modeling, analyzing, and supporting the process of online student knowledge sharing. These analysis techniques are used to train

a system to dynamically recognize (a) when students are having trouble learning the new concepts they share with each other and (b) why they are having trouble. In the second application, Interactive Multi-Media EXercises (IMMEX) Collaborative, a combination of iterative, nonlinear, machine learning algorithms is applied to identify latent classes of student problem-solving strategies. The approach is used to predict students' future behaviors within a scientific inquiry environment and provide targeted nonintrusive facilitation. The final section in this paper summarizes these analysis techniques and discusses how they might benefit other interdisciplinary areas.

A. INTRODUCTION TO APPLIED PROBABILISTIC CLASS ANALYSES

This section introduces the analysis methods and terminology that will be used in the remainder of this paper. The two artificial intelligence models discussed in this section and the next provide the foundation for the cognitive assessment and collaborative learning applications to follow. The goal in this paper is not to provide a comprehensive overview of probabilistic class analysis methods but, rather, to provide examples of the sorts of problems that seem to be amenable to latent class analysis (LCA) but for which more advanced methods, such as neural networks or HMMs, might provide a new perspective. This section begins with a simple example that lends itself to LCA and moves quickly into a discussion of more advanced statistical methods.

LCA has been used successfully to identify unobserved variables that explain the covariation (or nonindependence) within a known set of observed variables. The unobserved variable is termed *latent* because it is presumably unknown even though it might be hypothesized (McCutcheon, 1987). The result of applying this method should be a model in which the latent classes render the observed variables to be locally independent of each other.

The practical application of this method can be explained through a simple notional example from the area of defense analysis. Suppose our dataset describes blue (friendly force) team reports of red (enemy force) team activities. Each blue report would contain observed variables, such as activity type (e.g., weapon detonation, weapon emplacement, or movement of weapon components), activity time (e.g., early morning in June, late afternoon on Saturday), and location. Blue team reporting may also include observables that constrain the scope of the red team's available tactics, such as their disposition (e.g., strength of force, training, morale), equipment, weapons, terrain, and weather conditions.

We would see covariation among these observed variables and thus hypothesize that an unknown variable, such as red team strategy type (e.g., strategy of attrition or limited aims), explains why they chose a particular course of action and avoided an alternative. For example, an early morning explosion-based attack on the periphery of a small village may indicate a red team strategy that is ignorant of the ability of blue team radars to detect unusual activity in such areas with little or no "clutter." Late afternoon rush-hour activities in large cities involving the movement of weapons and components may be more representative of red team strategies that take advantage of the populous urban warfare terrain. Confirmatory LCA in this case offers the possibility that an unknown but hypothesized latent variable (e.g., red team strategy type) can explain the relations among the observed variables to the level of chance covariation (McCutcheon, 1987).

LCA uses observed data to estimate model parameters. The parameters are determined iteratively, commonly using an expectation maximization (EM) variant (see Baum, 1972; Baum, Petrie, Soules, & Weiss, 1970). They describe the likelihoods of the latent classes and the conditional response probabilities for the manifest variables within each latent class. For example, a conditional response probability parameter might describe the likelihood that the enemy who is employing a particular urban warfare strategy will emplace a weapon in the early morning at a particular location.

In the field of artificial intelligence, an array of nonlinear classification methods termed *unsupervised machine learning algorithms* offers approaches similar to LCA. The term *unsupervised* suggests that only the inputs to the model are given, and the goal is to discover (or *learn*, hence *machine learning*) the output distribution. In contrast, s*upervised* methods assume that the conditional output distribution is also given, and the goal is to find an optimal mapping between the input and output.

For neural network analysis, algorithms are available in both the supervised and unsupervised forms. From one perspective, models developed using neural network analysis are less constrained than those developed using variants of LCA because the latent class probabilities and conditional response probabilities do not need to be known or estimated. Instead, the class membership of each observed sample feature vector is estimated based on a learned, weighted transformation function. In its simplest form, the neural network is a nonlinear transformation function that maps a set of weighted input variables onto a number of latent classes. The weights on the input variables are estimated by incrementally adjusting them while minimizing the total sum-squared or mean-squared error. The estimation algorithm is typically a gradient descent derivative such as

back-propagation (Looney, 1997). The uniqueness of the method comes from its genesis in an early neurocognitive model of the human brain in which neurons in the brain either excite and fire (emit a 1) or do not fire (emit a 0) (McCulloch & Pitts, 1943). As such, the weighted sum of observed variables in a neural network is subject to a threshold function that filters the input and outputs a positive (1) or negative (0 or –1) response for each latent class. Continuing with this metaphor, because the threshold function activates the neural network nodes that represent the latent classes, it is commonly termed an *activation* function. This activation function may take many forms, the most common of which is a sigmoid, as shown in Figure 1. The sum, s, in the sigmoid activation function is given by a weighted sum of input (observed) variables, where b represents the bias (axis of symmetry for s) and α represents the growth rate (steepness of the curve).

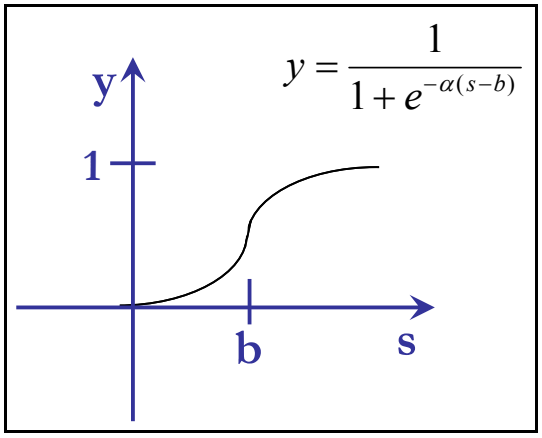

Figure 1. Sigmoid Activation Function

Neural networks have been used in cognitive and educational psychology applications to model student problem-solving strategies (Stevens, Ikeda, Casillas, Palacio-Cayetano, & Clyman, 1999) and the effectiveness of distributed collaborative interaction (Goodman, Linton, Gaimari, Hitzeman, Ross, & Zarrella, 2005). The input to the neural network in these cases consists of student activity (e.g., problem-solving actions, errors) or factors related to their conversation (e.g., type of speech act, presence of keywords, or certain punctuation marks). One application of neural networks combined with HMMs (described in the next section) is presented later in this paper.

LCA and neural network algorithms model systems of variables that remain constant over time and may not be appropriate for classes of problems whose variables describe processes that change dynamically. To illustrate, we revisit our earlier notional example of blue (friendly force) team responses to red (enemy force) team activities. Suppose we became aware that the red team was constantly changing its strategies to

accommodate changing conditions and to counteract blue team defensive strategies. Activities that the blue team may have easily detected and thwarted might subsequently become very difficult to ascertain. Such change in behavior over time is modeled through the use of longitudinal stochastic models.

The maturity of efficient maximum likelihood estimation algorithms enabled the development of effective longitudinal stochastic models (such as HMMs, described in the next section) that model and predict change over time. These models found some of their first applications in modeling learning and human behavior. For example, numerous applications of item response theory have successfully been shown to probabilistically relate students' individual characteristics to their responses to specific test items (De Boeck & Wilson, 2004). Soller, Martínez-Monés, Jermann, & Muehlenbrock (2005) reviewed a number of systems that apply artificial intelligence methodologies to support collaborative distance learning; however, the open issues, questions, and application possibilities still outnumber the research conducted thus far. The next section provides some technical background for the longitudinal stochastic latent class model that will be further developed through applications in the remainder of this paper.

B. MODELING STOCHASTIC CHANGE OVER TIME

1. Overview

A Markov chain is a useful tool for describing the way that samples taken at consecutive time intervals follow a representative path. A *mixed Markov model* is a mixture of a finite number of Markov chains. Mixed Markov models are thus restrictive in that each sample must be a member of one of the prescribed paths. The result of a mixed Markov analysis might describe the probability that a vector of samples taken from one subject is a member of a particular Markov chain describing that subject's behavior. For example, Langeheine & van der Pol (2002) described the utility of mixed Markov model analysis in modeling the rate of change-of-life satisfaction for a population across several years. The subjects fell into representative groups that were either satisfied or dissatisfied with their lives and continued to generally feel the same over time, or they fell into groups that changed from feeling generally dissatisfied to gaining some satisfaction or vice versa. The mixed Markov model described the likelihood that each subject was a member of each group (chain) and exhibited its corresponding behavior.

Mixed Markov models require that each sample be a member of one of the prescribed paths described by the transition probabilities. Such limitations can be

overrestrictive in longitudinal data analyses because individuals may change over time, and such models do not allow individuals to move between latent classes over time (Vermunt, 2007). The latent transition (or hidden) Markov model lessens the restrictions of mixed Markov models by allowing latent transitions, and the latent mixed Markov model provides for a mixture of Markov chains with latent transitions. Thus, the response probabilities and transition probabilities for the Markov chains in the model can change over time. In this way, we are able to model the way that the rate of change for each subject also changes over time as different temporal variables affect the way in which subjects respond to stimuli.

The next section presents a brief introduction to hidden Markov modeling to provide background and context for the applications described in the second half of this paper. A more complete formalization can be found in Rabiner (1989).

2. Hidden Markov Modeling

In this paper, latent transition and latent mixed Markov models will be referred to as hidden Markov models (or HMMs). The term *hidden* refers to the unobservable (latent) doubly stochastic process described by the latent transitions and the stochastic distribution of observations at each state. For example, observations might be classifications of different student problem-solving strategies with state transitions describing the likelihoods of transitioning from one general problem-solving strategy to another (e.g., on the next problem set or during the next term). In a collaborative distance learning environment, observations might be sequences of online chat between students, and state transitions might describe the communicative roles of students (e.g., facilitator, critic, peer tutor) or the effectiveness of the information sharing and knowledge construction.

HMMs can be used to perform three fundamental types of analyses:

1. Estimating a model that best characterizes a set of observations
2. Explaining sequences of observations, events, or behaviors in terms of latent class membership
3. Predicting the likelihood of future observations, events, or behaviors.

The model estimation (1)[1] is generally done first because the HMM that is the output of this step is subsequently used to perform analyses (2) and (3). The Baum-Welch

[1] The numbers (1), (2), and (3) in this paragraph refer to the list in the preceding paragraph.

or EM algorithms (Baum, 1972; Baum, Petrie, Soules, & Weiss, 1970) are commonly used for parameter estimation (1) and prediction (3), and the Viterbi (1967) algorithm is commonly used for revealing the most likely sequence of latent classes transited by a given observation sequence (2).

We begin by introducing the HMM terminology and notation. An HMM is specified by the set of parameters that describe the model's prior probabilities, state transition probabilities, and observation symbol probabilities. At any given time (t_1, t_2, and so forth), the model is understood to be in one of N states: $\{S_1, S_2, \ldots S_N\}$. The variable q_t denotes the state at time t, and the sequence of states traversed by the model is denoted by $Q = q_1, q_2, \ldots, q_T$.

The matrix of prior probabilities (π) describes the unconditional likelihood of each state before beginning the iterative process of parameter estimation known as HMM *training*. The prior probabilities of each state, q_i, are given by the initial state distribution, $\pi = \Pr[q_1 = S_i]$.

The *transition probabilities* describe the likelihoods of transiting from state i to state j, that is, $\Pr(q_j \mid q_i)$. These values are stored in matrix $\mathbf{A} = \{a_{ij}\}$, where

$$a_{ij} = \Pr[q_t = S_j \mid q_{t-1} = S_i].$$

The equation above states that the transition matrix describes the probabilities of the states q_t in the model, given that the previous state was $q_{t\text{-}1}$ (see Figure 2). The way in which the HMM stochastically transits through states over time enables it to model dynamic temporal processes, such as communication patterns or the shifting of problem-solving strategies over time.

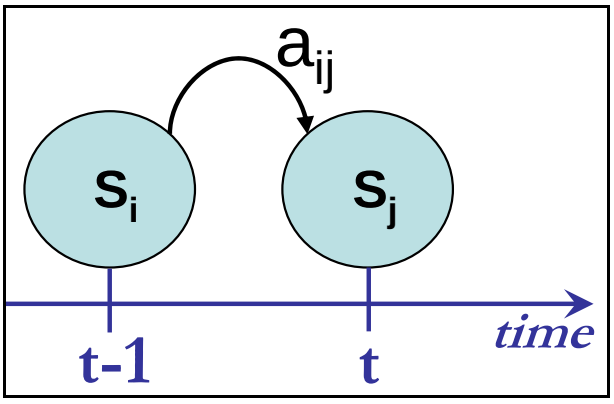

Figure 2. Illustration of Notation for One HMM State Transition

The last set of parameters describes **B**, the matrix of observation symbol probabilities for each state. Let O define an observation sequence, for example, a patient's changing mood over 6 months: O = {Anxious, Distressed, Sad, Scared, Relieved}. The observation symbol probability distribution describes the probabilities of each of the observation symbols, $O = O_1, O_2, \ldots, O_T$ for each of the states at each time t. The *observation symbol probability distribution* in state j is given by $\mathbf{B} = \{b_j(o_t)\}$, where

$$b_j(o_t) = \Pr[o_t = O_t \mid q_t = S_j].$$

O describes the set of all possible observation symbols.

In the first application described later, the observation symbols are given by online chat communication and problem-solving actions, and, in the second application, the observation symbols are given by the output of a neural network that describes student problem-solving strategies. Figure 3 illustrates an HMM for the notional defense analysis example that was described previously. Note how the temporal nature of the model accounts for how the unknown red team strategy might shift over time as soldiers are trained in different areas and as conditions change.

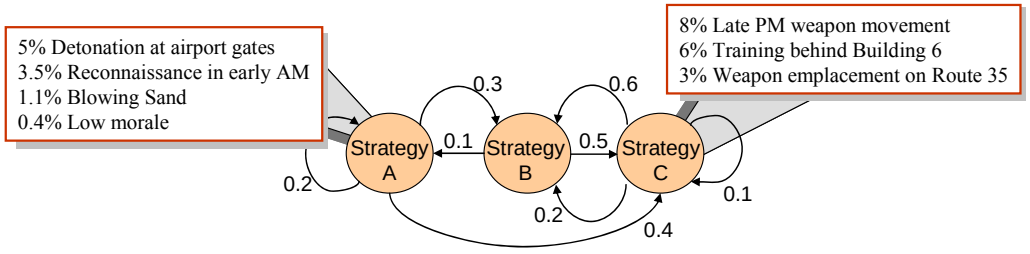

Figure 3. Depiction of HMM for a Notional Defense Analysis Example

Formally, if we let $\pi = \{\pi_i\}$ describe the initial state distribution, where $\pi_i = \Pr[q_1 = S_i]$, then an HMM (λ) can be fully described as

$$\lambda = (\mathbf{A}, \mathbf{B}, \pi),$$

where $\mathbf{A} = \{a_{ij}\}$ is the state transition matrix for the HMM and $\mathbf{B} = \{b_j(o_t)\}$ is the observation symbol probability distribution for each state j.

The Baum-Welch algorithm (Baum, 1972) is the EM algorithm for computing (learning) the maximum likelihood estimate of the HMM parameters, given samples of observation vectors. The E step of the algorithm provides the update rules for estimating the parameters in **A** (state probability distribution), and the M step describes the expected likelihood that the system will be in a given state and emit a particular observation in

B (observation probability distribution). An explanatory presentation of the full derivation of the Baum-Welch algorithm can be found in Bilmes (1998). Although the algorithm is not guaranteed to converge at the optimal solution (global maximum), it has been found to produce good results in practice based on local maxima (Rabiner, 1989).

After an HMM is estimated from a set of observations, it can be used to explain sequences of observations, events, or behaviors in terms of latent class membership, or it can be used to predict the likelihood of future observations, events, or behaviors. Because examples of both types of analysis are given later, we provide some technical background here for both algorithms. The reader who is more application-oriented may skip to the next section without loss of continuity.

Given an HMM, the Viterbi (1967) algorithm finds the most likely latent class (state) sequence for a given sequence of observations. This is the problem of finding the maximum $\Pr(Q \mid O, \lambda)$. For a given observation sequence, we describe the probability of the most likely state sequence at any time, *t*, as follows:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} \Pr[q_1, q_2, \ldots, q_t = i, O_1, O_2, \ldots, O_t \mid \lambda) .$$

To identify the state sequence that produces the maximum likelihood result for the entire observation sequence, the algorithm saves the argument that produces the best result at each time and state for $\delta_{t+1}(j)$:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(O_{t+1}) .$$

Observation sequences that begin the same but diverge over time may produce very different state sequences because as observations accumulate, the HMM recalculates the most likely state sequence. Thus, it is not always possible to know what state the HMM is in (it is *hidden*).

The forward-backward procedure is used to estimate the likelihood of an observation sequence, given an HMM (Rabiner, 1989). This likelihood is denoted $\Pr(O \mid \lambda)$. Let $\alpha_t(i) = \Pr(O_1, O_2, ..., O_t, q_t = S_i \mid \lambda$. The variable $\alpha_t(i)$ is called the forward variable and describes the probability of a partial observation sequence (up until time *t*), given model λ. In the first step, we initialize $\alpha_t(i) : \alpha_1(i) = \pi_i b_i(O_1)$. This initializes the forward variable as the joint probability of state S_i and the initial observation O_1. The second step (induction) is given by the following equation, in which *N* denotes the number of states in the HMM:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) .$$

The sum, $\sum_{i=1}^{N} \alpha_t(i) a_{ij}$, describes the probability of the joint event in which O_1, O_2, …, O_t are observed, the state at time t is S_i, and the state S_j is reached at time $t+1$. In other words, it is the probability of being in state S_j at time $t+1$, accounting for all the accompanying previous partial observations. Then, $\alpha_{t+1}(j)$ can be determined by multiplying this value by $b_j(O_{t+1})$.

The final estimated value of $\Pr(O \mid \lambda)$ is then given by summing over the terminal values:

$$\Pr(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) .$$

This completes our technical discussion of stochastic temporal class analysis methods. Additional introductory information on similar types of methods can be found in Rabiner (1989), Looney (1997), McCutcheon (1987), and Bilmes (1998). The next section discusses applications of these methods to the areas of collaborative learning and longitudinal cognitive assessment.

C. APPLICATIONS OF PROBABILISTIC SEQUENTIAL CLASS ANALYSIS TO EDUCATIONAL ASSESSMENT

Applications of probabilistic class analysis to psychology and education are copious in the literature, with references reaching back as early as the 1950s (Miller, 1952). Examples include Greeno's (1967) Markov chain models of paired-associate learning (also see Greeno & Steiner, 1964), Kintsch & Morris' (1965) Markovian models of recall and recognition, Brainerd's (1979) models of conservation learning, and Wickens' (1982) stochastic models of short- and long-term memory. This section describes two different applications in which longitudinal stochastic class analysis methods are used to assist an instructor or online coach in assessing and mediating online student interaction with the aim of improving the quality of students' distance learning experiences.

The first application, EPSILON, applied a combination of hidden Markov modeling and MDS for modeling, analyzing, and supporting the process of online student knowledge sharing. These analysis techniques were used to train a system to recognize

dynamically (a) when students are having trouble learning the new concepts they share with each other and (b) why they are having trouble.

1. Encouraging Positive Social Interaction while Learning ON-Line (EPSILON)

The EPSILON project was motivated by the rapid advance of networked collaborative and distance learning technology that has enabled universities and corporations to reach across time and space barriers to educate learners. While this technology has removed many of the traditional constraints surrounding when, how, and what can be learned, the quality of distance learning still falls behind the standards set by structured face-to-face learning. The nature of the communication medium itself is partly responsible because supporting and mediating large numbers of online collaborative learning teams would require online instructors to spend an extraordinary amount of time reviewing chat, email, and newsgroup discussions. The EPSILON effort aimed to demonstrate how a computer might assist an online instructor in assessing the effectiveness of students' collaborative learning interaction. The prerequisite for this effort involved identifying and understanding the processes involved in distributed collaboration and determining the support needed to facilitate and enhance these processes (Soller & Lesgold, in press). For the remainder of this section, we focus on just one of these processes: the process of information sharing.

From a theoretical standpoint, a group's ability to share, understand, and construct new knowledge is an important predictor of the value of the distributed collaborative learning experience. The effectiveness of knowledge construction depends on the participants' evolving knowledge bases and the group's ability to share and assimilate the bits of knowledge necessary to construct new knowledge. As information is shared and assimilated into the group's thinking process, group members evolve and develop a common understanding. From an intuitive standpoint, the knowledge that group members bring to bear on the problem and how this knowledge is shared, understood, and further developed (or not) ultimately shape both the process and the product of the collaboration. This section shows how some of the procedures described in the first part of this paper can be applied to analyze the process of knowledge sharing during collaborative distance learning activities.

2. Experimental Design

The study (Soller, 2004) was designed in the style of traditional Hidden Profile studies in social psychology (Lavery, Franz, Winquist, & Larson, 1999; Mennecke, 1997;

Stasser, 1999), which are specifically oriented to evaluate the effect of information sharing on group performance. Hidden Profile studies require that the knowledge needed to perform the task be divided among group members, such that each member's knowledge is incomplete before the group session begins. The group task is designed so that it cannot be successfully completed until all members share their unique knowledge. Group performance is typically measured by counting the number of individual knowledge elements that surface during group discussion and evaluating the group's solution, which is dependent on these elements. Although some studies do suggest that the quality and quantity of unique information shared by group members is a significant predictor of the quality of the group decision (Hollingshead, 1996; Winquist & Larson, 1998), other studies have historically and consistently shown that group members are not likely to discover their teammates' hidden profiles (Lavery et al., 1999; Stasser, 1999). Group members tend to focus on information that they share in common and tend not to share and discuss information they uniquely possess. The aim of the study described in this section was to identify the various ways that group members can effectively share information with each other and the various ways that they can experience knowledge-sharing breakdowns. The results of this analysis could serve to inform and advise an instructor in selecting an appropriate facilitation strategy.

Twelve groups of three participants each participated in the study; however, the sample size was related less to the number of participants than to the amount of conversation about unique knowledge elements. All the subjects [except for two technical staff members from a participating Federally Funded Research and Development Center (FFRDC)] were undergraduates or first-year graduate students majoring in the physical sciences or engineering, and most were not experienced in the domain of Object-Oriented Analysis and Design (OOA&D). Each group was asked to solve one problem, using a collaborative graphical OOA&D workspace while communicating through a structured chat interface. The chat interface contained sets of *sentence openers* (e.g., "I think," "I agree because") organized in intuitive categories (e.g., Inform, Request, or Acknowledge) that the students used to indicate (to the system) the general intentions underlying their chat contributions. After a brief training and practice period, the students were assigned to separate rooms, given *individual knowledge elements* (described next), and took a pre-test. More detailed information about the tool and experimental design can be found in Soller (2004).

As in the Hidden Profile studies described previously, the key knowledge elements needed to solve the OOA&D problems were distributed among the three students

in each group before the problem-solving session started. These three individual knowledge elements represented conceptual elements, such as "Attributes common to a group of subclasses should be attached to the superclass and will be inherited by each subclass." The distribution of knowledge elements was intended to reflect the natural distribution of knowledge among people with different expertise. While this experimental design does not preclude situations in which one student may know a concept while another has a deeply rooted misconception, it is perhaps less reflective of such situations because the student would need to have the misconception prior to the study.

The students were pre-tested on all three knowledge elements before the problem-solving session and post-tested afterward. It was expected that the student given knowledge element #1 would get only pre-test question #1 correct, the student given knowledge element #2 would get only pre-test question #2 correct, and the student given knowledge element #3 would get only pre-test question #3 correct. To ensure that each student understood his unique knowledge element, an experimenter reviewed the pre-test problem pertaining to each student's knowledge element before the group exercise. During the on-line problem-solving session that followed, the software automatically logged the students' conversation and actions. After the problem-solving session, the subjects completed a post-test, which assessed the extent to which the students learned the two knowledge elements from their peers.

The problem-solving session logs were segmented by hand to extract the segments in which the students shared their unique knowledge elements. A total of 29 *knowledge-sharing episodes* were manually identified, and each was classified as either an effective knowledge-sharing episode or a knowledge-sharing breakdown.[2] The manual segmentation procedure involved identifying the main topic of conversation by considering both the student dialog and the workspace actions (such as creating or augmenting a new graphical OOA&D object), and the classification was based on an examination of the pre- and post-test scores. A sequence was considered a knowledge-sharing breakdown if the knowledge element was discussed during the episode, but none of the receiving students demonstrated mastery of the concept on the post test. The sequence was considered effective if at least one of the participants learned the concept during the session. The 29 knowledge-sharing episodes varied in length from 5 to 49 contributions and contained both conversational elements and OOA&D actions.

---

[2] Ten sequences were identified as effective knowledge-sharing episodes, and 19 sequences were identified as examples of knowledge-sharing breakdowns.

The top part of Figure 4 shows an example of one such episode. The italicized *sentence openers* in the figure were used by the system to automatically code the utterances' subskills and attributes, which formed the basis for the HMM analysis. The bottom part of Figure 4 shows the corresponding sequence that was used to train the HMMs to analyze and classify new instances of knowledge sharing (described in the next section).

Student	Subskill	Attribute	Text Chat
A	Request	Opinion	*Do you think* we need a discriminator for the car ownership
A	*<Begins to construct a discriminator on the Collaborative Workspace>*		
C	Discuss	Doubt	*I'm not so sure*
B	Request	Elaboration	*Can you tell me more* about what a discriminator is
C	Discuss	Agree	*Yes, I agree* because I myself am not so sure as to what its function is
A	Inform	Explain	*Let me explain it this way* - A car can be owned by a person , a company or a bank. I think ownership type is the discrinator.

Actual HMM Training Sequence
A-Request-Opinion
A-OOA&D-Action
C-Discuss-Doubt
B-Request-Elaboration
C-Discuss-Agree
A-Inform-Explain

Figure 4. Example of a Logged Knowledge-Sharing Episode, Showing System-Coded Subskills, Attributes, and a Corresponding HMM Training Sequence

In a preliminary analysis, a prototype HMM classifier was able to determine (with 100 percent accuracy) which of the three students played the role of knowledge sharer during the identified knowledge-sharing episodes (Soller & Lesgold, in press). This analysis was performed because, if successful, it would allow the system to assign a special set of tags to the contributions of the knowledge sharer. In Figure 4, for example, the tags reserved for the knowledge sharer's contributions begin with the code "A-". The contributions of other two students were arbitrarily assigned the codes "B-" and "C-". Differentiating the roles of the knowledge sharer and recipients was thought to facilitate the system's assessment of the episodes' effectiveness.

3. Hidden Markov Modeling of Knowledge Sharing

Two 5-state HMMs were trained using the MATLAB HMM Toolbox (available from Kevin Murphy at http://www.ai.mit.edu/~murphyk/Software/HMM/hmm.html).

Five states were chosen because preliminary analysis results showed that 3-, 4-, and 6-state HMMs produced less favorable (although somewhat similar) results and performance declined with seven or more states. The first HMM was generated, as described previously, using only the 10 sequences of effective knowledge-sharing interaction (this will be termed the effective HMM), and the second HMM was generated using only the 19 sequences of ineffective knowledge sharing, or knowledge-sharing breakdowns (the ineffective HMM). The ability of the HMMs to effectively model the behaviors exemplified by the observation sequences was tested using a modified "take-2-out" 58-fold cross-validation approach. Each of the observation sequences was replicated with actors B and C swapped so that the analysis would not reflect idiosyncrasies in the labeling of participants B and C. This resulted in a total of 58 episodes (or 29 pairs of episodes). Then, each test sequence and its B-C swapped pair were removed from the training set and tested against the two HMMs (representing effective and ineffective interaction) that were trained using the other 56 episodes.

Testing the models involved computing the probability of a new knowledge-sharing sequence—one that is not used for training—given both models. The output given the effective HMM described the probability that the new test sequence was effective (as defined by the training examples), and the output given the ineffective HMM described the probability that the test sequence was ineffective (see Figure 5). The test sequence was then classified as effective if it had a higher path probability through the effective HMM or ineffective if its path probability through the ineffective HMM was higher. Procedures similar to this have been used successfully in other domains, such as gesture recognition (Yang, Xu, & Chen, 1997) and the classification of international events (Schrodt, 2000).

It is not necessarily intuitive that two probabilities, generated by models trained from different data sets, are comparable or even indicative of the effectiveness of a test sequence. The procedure discussed previously described how to obtain $\Pr(S \mid \lambda)$, the probability of a test sequence given an HMM. If we would like to test the effectiveness of a sequence, we need to compare $\Pr(S \mid \lambda_{\text{eff}})$ to $\Pr(S \mid \lambda_{\text{ineff}})$. As long as the models are initially seeded using the same constraints, we can obtain the same result by comparing $\Pr(\lambda_{\text{eff}} \mid S)$ to $\Pr(\lambda_{\text{ineff}} \mid S)$. Formally, we can compute $\Pr(\lambda \mid S)$ by Bayes' Rule:

$$\Pr(\lambda \mid S) = \frac{\Pr(S \mid \lambda)\Pr(\lambda)}{\Pr(S)}.$$

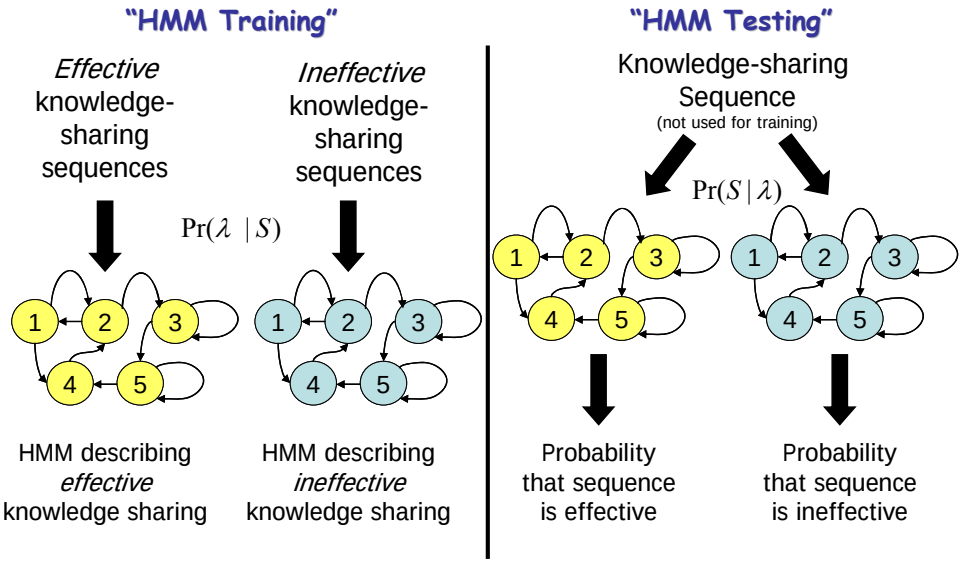

Figure 5. Schematic of Procedure for Training and Testing the HMMs To Assess the Effectiveness of Student Knowledge Sharing

In comparing $\Pr(\lambda_{\text{eff}} \mid S)$ to $\Pr(\lambda_{\text{ineff}} \mid S)$, the probability of the test sequence, $\Pr(S)$, is a constant because the same test sequence is run through both models. It can, therefore, be eliminated. This leaves us with the comparison of $\Pr(S \mid \lambda_{\text{eff}})\Pr(\lambda_{\text{eff}})$ to $\Pr(S \mid \lambda_{\text{ineff}})\Pr(\lambda_{\text{ineff}})$. Because the models' λ_{eff} and λ_{ineff} are also constants across all the test cases and do not differ statistically significantly ($p = 0.65$), they too can be eliminated, leaving us with $\Pr(\lambda \mid S) \cong \Pr(S \mid \lambda)$. The p statistic, obtained through a Kolmogorov-Smirnov test, tells us that the distributions of transition probabilities in the two models do not differ significantly (Fisher & van Belle, 1993). Since the HMMs remain constant for all the test cases, it is reasonable to perform relative comparisons of $\Pr(\lambda_{\text{eff}} \mid S)$ and $\Pr(\lambda_{\text{ineff}} \mid S)$, although the absolute magnitudes of the differences between the models may not be significant. In summary, it may be more computationally intuitive to think of the analysis that follows as a process of comparing two HMMs—one effective and one ineffective—and determining which model best matches a given test sequence. However, because this is essentially the same as the more conventional terminology in which we calculate the likelihood of a sequence, given a model, we have adopted the latter form.

As seen in Figure 6, 16 of the 20 effective knowledge-sharing sequences were correctly classified by the effective HMM, and 27 of the 38 ineffective sequences were correctly classified by the HMM modeling knowledge-sharing breakdowns. Overall, the HMM approach produced an accuracy of 74.14 percent, almost 25 percent above the baseline. The baseline comparison for this analysis is chance, or 50 percent, because

	Effective HMM	Ineffective HMM	
Effective Test Sequences	16	4	0.8
Ineffective Test Sequences	11	27	0.7

Figure 6. Results of HMM Analysis

there was a 0.5 chance of arbitrarily classifying a given test sequence as effective or ineffective and the sample size was not large enough to establish a reliable frequency baseline.

This analysis showed that HMMs are useful for identifying when group members are effectively sharing information and when they are experiencing knowledge-sharing breakdowns. A system based on this analysis alone could offer support and guidance about 74 percent of the time the students need it, which is better than guessing when students are having trouble or basing intervention solely on students' requests for help. The next step is to determine why students may be having trouble so that appropriate facilitation methods can be identified and tested. The following section takes a closer look at the differences between the effective and ineffective sequences in order to understand the qualitative differences.

4. Multidimensional Scaling (MDS) of Hidden Markov Model (HMM) Likelihoods

An HMM clustering approach (Juang & Rabiner, 1985; Smyth, 1997) was used to develop generalized models of effective knowledge sharing and breakdowns in knowledge sharing. The approach involved a combination of hidden Markov modeling, MDS (Shepard & Arabie, 1979), and a self-organizing clustering routine. In the first step of this approach, each of the knowledge-sharing episodes was used to train one HMM (in the traditional manner). This resulted in 29×2 paired HMMs, each pair representing a generalization of a particular knowledge-sharing behavior.

Formally, each sequence, S_j, $1 \leq j \leq N$, was used to train one HMM, M_i, $1 \leq i \leq N$, $i = j$. For the effective case, $N_{\text{eff}} = 20$, and, for the ineffective case, $N_{\text{ineff}} = 38$. Then, the

log-likelihood of each sequence, S_j, given each of the HMMs, M_i, was calculated via the standard HMM testing procedure. This resulted in two matrices, one describing the log-likelihoods of the effective sequences given the effective models, $loglik_{\text{eff}}(S_j \mid M_i)$ and another describing the log-likelihoods of the ineffective sequences given the ineffective models, $loglik_{\text{ineff}}(S_j \mid M_i)$. The columns of these matrices described the likelihood of each of the sequences given a particular model, M_i; hence, similar HMMs should produce similar column vectors, which we will call likelihood vectors. Given this observation, it would make sense to cluster these column vectors and identify models that were most similar as model groupings. Traditional hierarchical clustering approaches, however, did not work well because the outlier data points caused the generation of single clusters from singleton data points. To deal with this problem, the data were analyzed using an MDS procedure in which the likelihood vectors were positioned in a multidimensional space that was divided into regions describing the groups of HMMs (Kruskal & Wish, 1978; Shepard, 1980).

The MDS approach was attractive for this research because each of the groupings found in the multidimensional space described a particular way in which group members effectively share new knowledge with each other or experience breakdowns while attempting to share new knowledge with each other. The full algorithm to perform the MDS of HMM likelihoods is described in Soller (2004). Briefly, the standard MDS procedure was applied to the HMM log-likelihood matrices, such that $loglik(S_j \mid M_i) \rightarrow D_{ji}$, where $D_{ji} = d(L_{Mj}, L_{Mi})$ describes the Euclidean distance between the N HMM likelihood vectors in a three-dimensional (3-D) space (Kruskal & Wish, 1978). The likelihood vectors were then assigned to groups based on the closeness of the data points in the MDS scaled configurations.

The groups of scaled HMM likelihood vectors were verified using an iterative, self-organizing data analysis technique (ISODATA) along with a maximum distance threshold criteria (Looney, 1997). The maximum distance threshold enabled the algorithm to ignore those points that were too far away from any of the established clusters. The dataset that was analyzed was small compared to the number of different ways students can share new knowledge with each other. Even though some of the models in the dataset may represent single examples of certain types of interaction, only those models for which several examples exist can be reliably classified and analyzed. The additional maximum distance threshold criteria ensured that those models represented by only a single example would not be forced into a cluster.

Three groups of effective HMMs and four groups of ineffective HMMs were discovered (see Figure 7). Each grouping was compared to a qualitative analysis of the student activity in each of the groups. The episodes were first summarized blindly, without knowledge of the groupings. Then, the summarized episodes were compared to the clusters that were found computationally. The remainder of this section describes the sort of interaction that occurs when students attempt to share new knowledge with each other, as suggested by the computational procedure.

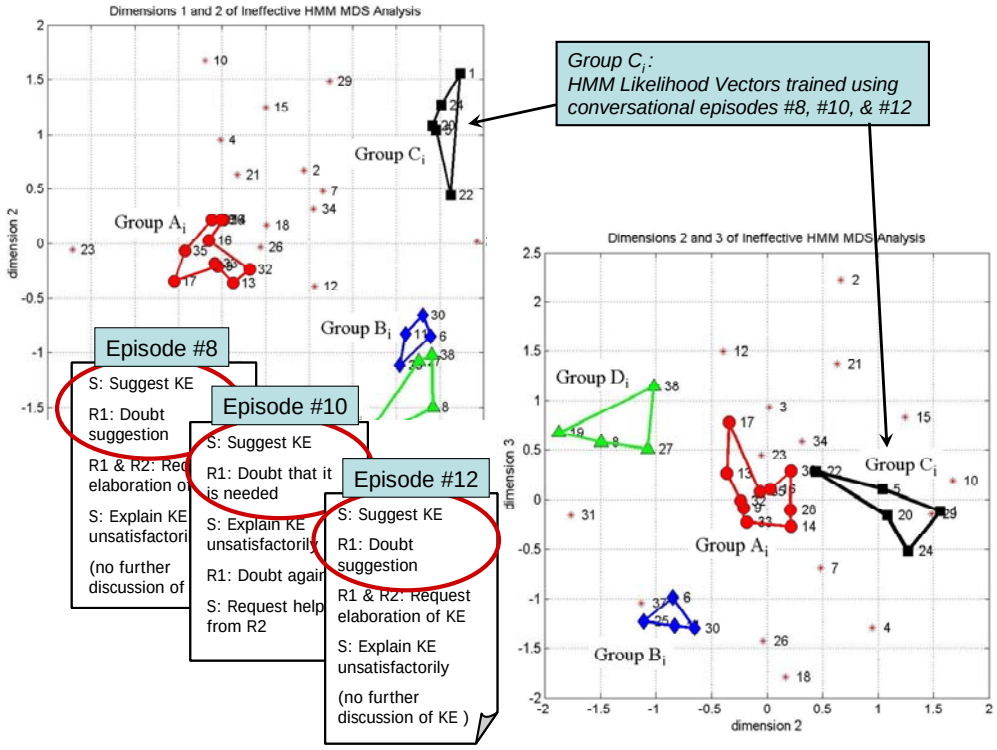

Figure 7. HMM Likelihood Vector Clustering for Knowledge-Sharing Breakdown Groupings

Figure 8 shows the four generalized models that were found from the groups of ineffective likelihood vectors (A_i, B_i, C_i, and D_i). In the first group (A_i), the sharer (student A) first proposes that the group discuss his knowledge element. The sharer then proceeds to either explain the knowledge element or gives instructions to one of the receivers (students B or C) for applying the knowledge element concept to the exercise. The episode closes when the receiver(s) simply acknowledges or requests confirmation of his actions. In the second group (B_i), the sharer first attempts to explain his knowledge element. This act is followed by only acknowledgement, and no further explanation. In the third group (C_i), the sharer proposes his knowledge element. This act is followed by

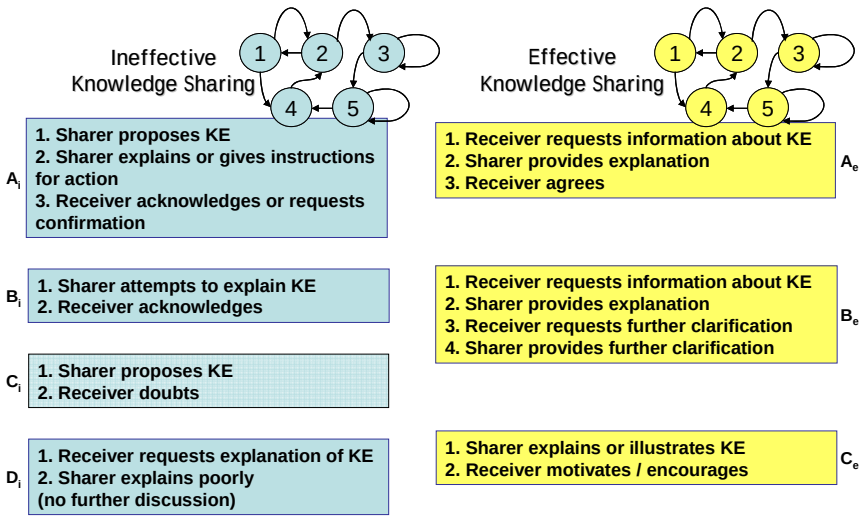

Figure 8. Summarized HMM "Learned" Knowledge-Sharing Examples

doubt on the part of the receivers. The blindly summarized knowledge-sharing episodes for group C_i and the corresponding likelihood vector grouping in the multidimensional space are illustrated in Figure 7. In the fourth group (D_i), one of the receivers first requests an explanation of one of the knowledge elements, after which the sharer explains his knowledge element poorly, ending the discussion on the knowledge element [see Soller (2004) for examples of knowledge-sharing breakdowns].

Figure 8 also shows the three generalized models that were found from the groups of effective examples (A_e, B_e, and C_e). Generally, the discussions in which students effectively shared and learned each other's knowledge elements were marked by questioning, explanation, agreement, and motivation, whereas the discussions in which the students experienced breakdowns in knowledge sharing were marked by poor (inaccurate or incomplete) explanations, instructions for action, doubt, and acknowledgement.

This section described a longitudinal stochastic analysis approach that combines HMM and MDS with a threshold-based clustering method. The approach provided insight into the various ways that students can share knowledge effectively and the various ways that students can have trouble sharing new knowledge. The analysis illustrated how effective knowledge-sharing discussions were markedly different from discussions in which the students experienced knowledge-sharing breakdowns. The results of this analysis could serve to inform and advise an instructor in selecting an appropriate facilitation strategy. The next section describes an application in which a different hybrid combination of longitudinal stochastic methods was used to model and assess cognitive development.

D. INTERACTIVE MULTI-MEDIA EXERCISES (IMMEX) COLLABORATIVE

IMMEX™ is a Web-based multimedia scientific learning environment that combines iterative nonlinear machine learning algorithms to identify latent classes of student problem-solving strategies. The single-user version, which was developed at the University of California, Los Angeles, has been used in science classes across middle and high schools, universities, and medical schools in the United States over the past 12 years and has logged over 250,000 student problem-solving performances (Stevens & Palacio-Cayetano, 2003). A rich portfolio of over 100 problem sets in various disciplines has been developed and is available online at http://www.immex.ucla.edu.

The IMMEX Collaborative (see Figure 9), which was augmented at the University of Trento, Italy, also includes general-purpose collaborative Web navigation and synchronization facilities and a structured chat interface (Ronchetti, Gerosa, Giordani, Soller, & Stevens, 2005). The IMMEX Collaborative environment is designed to help groups of students learn how to articulate hypotheses to each other (through a structured chat interface) and analyze laboratory tests while solving real-world problems. For instance, chemistry students learn how to discern the composition of unknown substances resulting from a chemical spill to determine if they are dangerous. The students use scientific inquiry skills to frame the problem, judge what information is relevant, plan a search strategy, select the appropriate physical and chemical tests to solve the problem (e.g., litmus, conductivity), and eventually reach a decision that demonstrates understanding. As the students work through the problems, the system logs their chemical and physical test selections, browser navigation actions, and chats. These actions then serve as the input vectors to self-organizing artificial neural networks (Kohonen, 2001) that are trained to recognize student problem-solving strategies.

The strategies students use to solve scientific inquiry problems, in which they must search for and evaluate the quality of information, draw inferences, and make quality decisions, provide evidence of their knowledge and understanding of the domain. In this section, we show the utility of artificial intelligence methods, in particular neural networks and HMMs, for automatically identifying students' individual problem-solving strategies and predicting their future strategies. If we can determine whether a student is likely to continue applying an inefficient problem-solving strategy, we may be able to determine whether the student will likely need help and guidance in the near future. Help

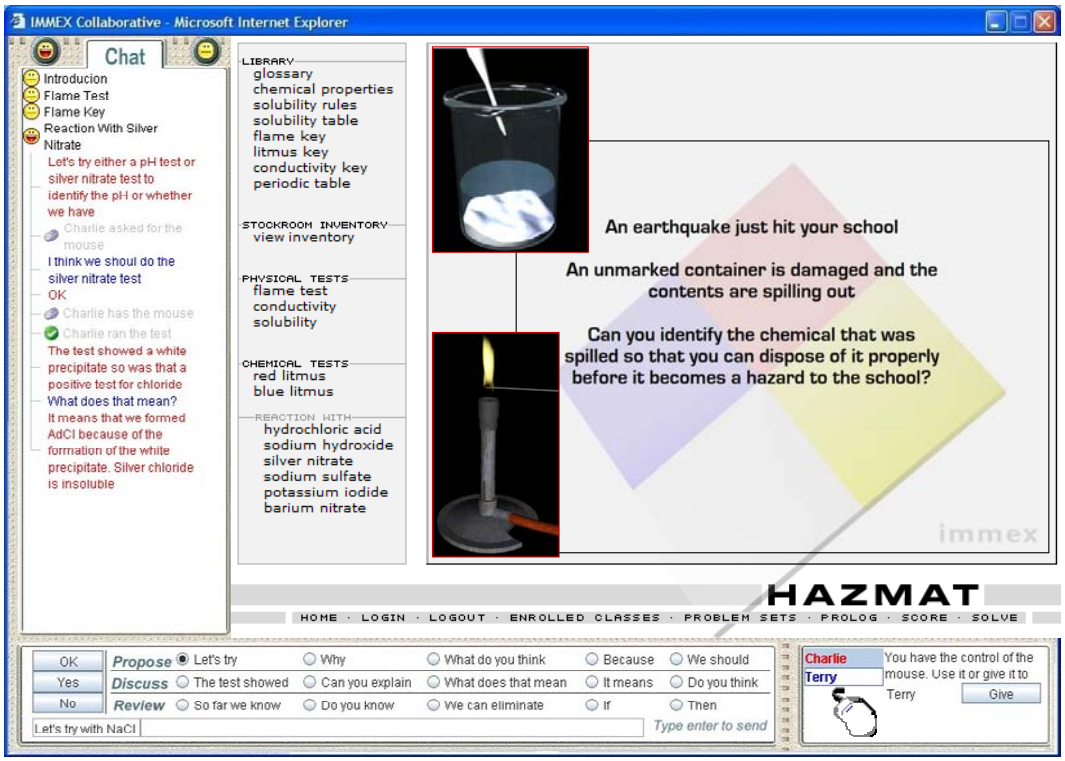

Figure 9. IMMEX Collaborative Interface

might be provided through direct intervention by a teacher or a computer-based coach or through indirect intervention by strategically setting up and mediating peer collaboration situations.

1. Item Response Theory (IRT) Modeling of Student Ability and Item Difficulty

Students provide evidence of their problem-solving strategies through the patterns of actions that they take when confronted with problems of varying levels of difficulty. The first step in developing metrics to assess student ability and problem-solving strategy development was to have students perform multiple problems that vary in difficulty. Estimates of their ability were initially obtained through IRT analyses, which describe the relative difficulty of problems and abilities of students. IRT relates characteristics of items (item parameters) and characteristics of individuals (latent traits) to the probability of a positive response (e.g., solving a case). Unlike classical test theory item statistics, which depend fundamentally on the subset of items and persons examined, IRT item and person parameters are invariant. This makes it possible to examine the contribution of items individually as they are added and removed from a test. It also allows researchers to conduct rigorous tests of measurement equivalence across experimental groups.

Using IRT, pooled student data were used to obtain a proficiency estimate for each student based on whether he solved each problem. The Winsteps program (Linacre, 2004) was used to compute proficiency scores and item difficulty estimates. Using the one-parameter logistic (1-pl) model as well as the two-parameter logistic (2-pl) model, Winsteps scales both the items and the individual examinees on the same logit (log-odds) scale:

$$\log\left[\frac{\Pr(x_j = 1)]}{\Pr(x_j = 0)}\right] = \theta_s - b_j .$$

The overall θ_s is an estimated proficiency based on the number of correctly answered items in a set. The higher the student ability, θ, the higher the probability of getting a more difficult item correct. The item difficulties, b_j, are the difficulty estimates of each item. The IMMEX case item difficulties were determined by IRT analysis of 28,878 student performances. Cases included a variety of acids, bases, and compounds, and the ability measures showed that the problem set presented an appropriate range of difficulties to provide good estimates of student ability.

The IRT analysis only estimated a minimal amount of information about the students' cognitive thought processes because item score (a coarse measure) was all that was used; however, it provided the necessary foundation for the follow-on neural network analysis (described in the next section), which performs a more fine-grained analysis of students' cognition and problem solving.

2. Neural Network Modeling of Problem-Solving Strategies

Statistics for over 5,000 individual problem-solving performances collected by the IMMEX system were used to train competitive, Self-Organizing Maps (SOMs) (Kohonen, 2001). A SOM is a type of unsupervised neural network that learns to group similar observation vectors in such a way that the nodes physically near each other respond similarly to like input vectors (Kohonen, 2001). In our case, the neural network observation (input) vectors described sequences of individual student actions during problem solving (e.g., Run_Blue_Litmus_Test, Study_Periodic_Table, Reaction_with_Silver_Nitrate). The result of the neural network training was a topological ordering of neural network nodes according to the structure of the data, such that the distance between the nodes described the similarity of the students' problem-solving strategies. For example, the neural networks identified situations in which students applied ineffective strategies (e.g., running a large number of chemical and physical tests or not

consulting the glossaries and background information) and effective strategies (e.g., balancing test selection with searching for background information). Other domain-specific, problem-specific strategies included repeatedly selecting specific tests (e.g., flame or litmus tests) when presented with compounds involving hydroxides (Stevens, Soller, Cooper, & Sprang, 2004). From a statistical perspective, nonlinear SOMs are similar to nonlinear *k*-means clustering variants with constrained topologies.

The resulting SOM took the form of a 36-node neural network, derived from the 5,284 performances of university and high school chemistry students, that described the 36 different problem-solving strategies used by the students. Each node of the network was represented by a histogram showing the frequency of items selected by students (see Figure 10). For example, 22 tests were related to Background Information (items 1–9), Flame Tests, Solubility, and Conductivity (items 10–12), Litmus tests (items 13, 14), Acid and Base Reactivity (items 15, 16), and Precipitation Reactions (items 17–22).

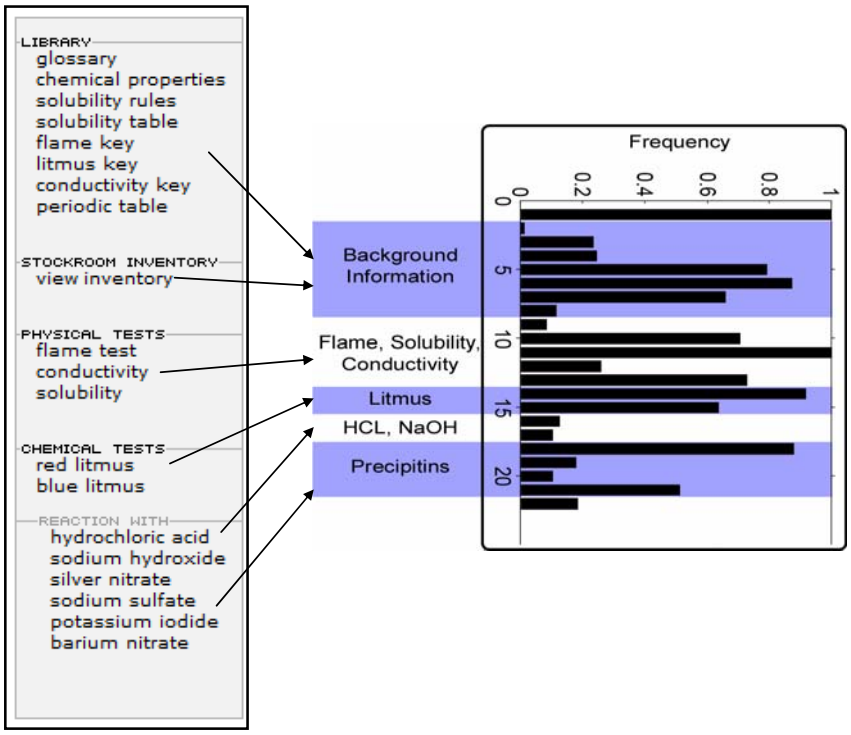

Figure 10. One Neural Network Node Describing the Frequency of Items Selected by Students at That Node

Choices regarding the number of nodes and the different architectures, neighborhoods, and training parameters have been described previously (Stevens, Wang, & Lopo, 1996). The 36 neural network nodes are represented by a 6×6 grid of 36 graphs (see

Figure 11). The nodes are numbered 1 through 36 left-to-right and top-to-bottom. Forexample, the top row is comprised of nodes 1 through 6. As the neural network is iteratively trained, the performances are automatically grouped into these 36 nodes so that each node represents a different generalized subset of the population. In this case, each subset describes a different problem-solving strategy. These 36 classifications are observable descriptive classes that can serve as input to a test-level scoring process or linked to other measures of student achievement. They can also be used to construct immediate or delayed feedback to the student or aggregated cognitive statistics for the instructor.

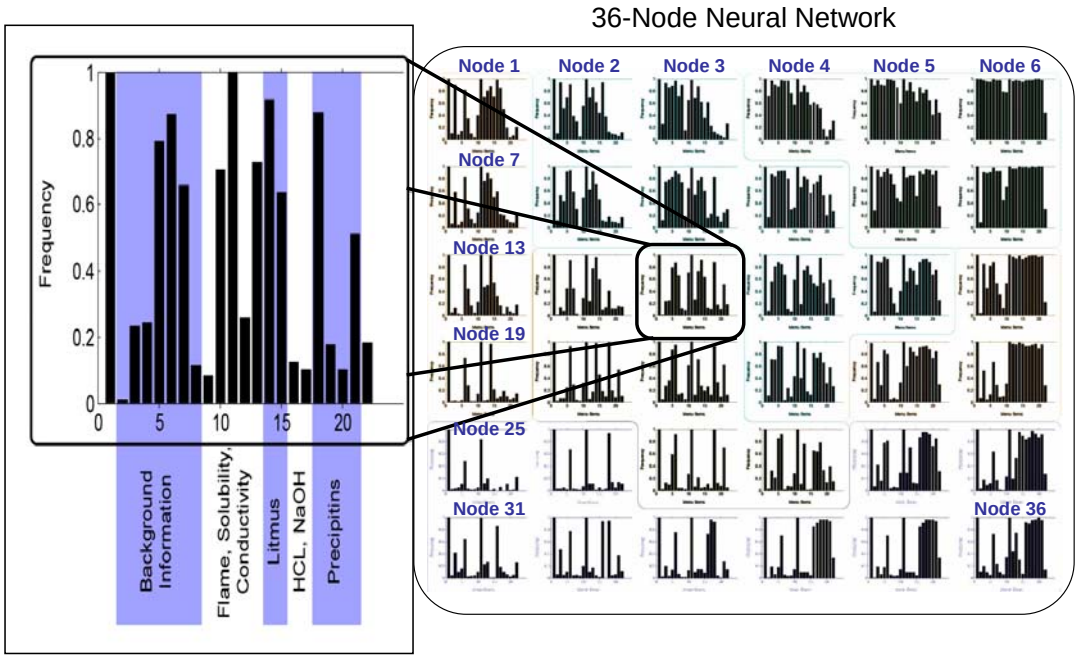

Figure 11. A Neural Network Showing the 36 Nodes, Each Describing a Different Subset of the Population

Many of the student performances that were grouped together at a particular node represent problem-solving strategies adopted by students who always selected the same tests (i.e., those with a frequency of 1). For instance, all Node 15 performances shown in the left-hand side of Figure 11 contain the items 1 (Prologue) and 11 (Flame Test). Items 5, 6, 10, 13, 14, 15 and 18 have a selection frequency of 60–90 percent, meaning that any individual student performance that falls within that node would most likely contain some of those items. Items with a selection frequency of 10–30 percent were regarded more as background noise than as significant contributors to the strategy represented by that node.

The topology of the trained neural network provides information about the variety of different strategic approaches that students apply in solving IMMEX problems. First, it is not surprising that a topology is developed based on the quantity of items that students select. The upper right hand of the map (nodes 6, 12) represents strategies in which a large number of tests are being ordered, whereas the lower left hand of the map (nodes 25, 31) contains clusters of strategies where few tests are being ordered. There are also differences that reveal the quality of information that students use to solve the problems. Nodes situated in the lower right hand corner of Figure 11 (nodes 29, 30, 34, 35, 36) represent strategies in which students selected a large number of items but no longer needed to reference the Background Information (items 1–9).

Each neural network node is associated with a corresponding solution frequency. The node's solution frequency describes the percentage of students at that node who successfully solved the problem. By linking the solution frequency to each of the neural network nodes, an indication of the efficiency of the different strategies can be obtained. Figure 12 shows the grayscale values for these nodal solution frequencies overlaid on the 36-node neural network map. The darker shades indicate lower solution frequencies.

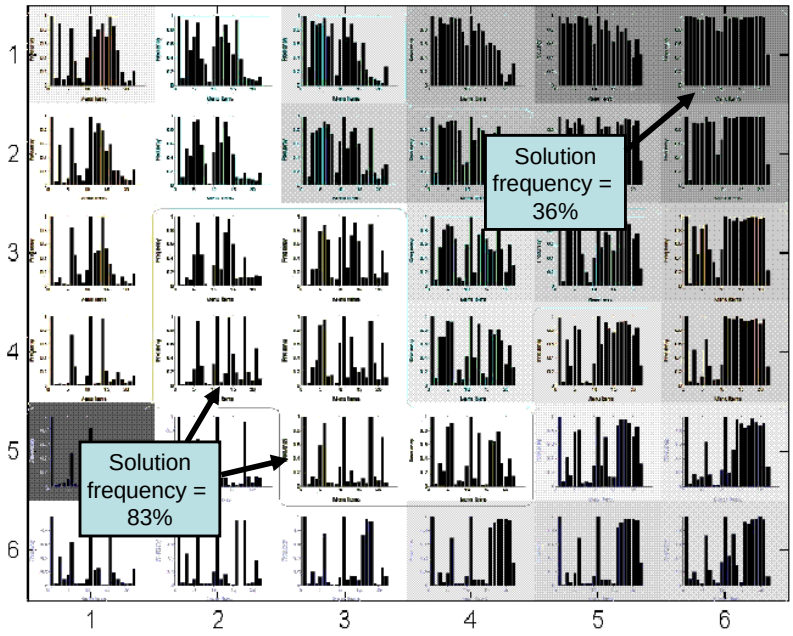

Figure 12. Solution Frequencies Overlaid on the 36-Node Neural Network Map

The figure shows that ordering all tests (nodes 5 and 6, upper right) or very few tests (node 25, lower left) are not efficient strategies. Effective strategies with the highest solution frequency were, for the most part, reflected by a balance of selecting background

and test items. These were best visualized in the lower-left hand corner and the middle of the topology map and are exemplified by students making particularly useful associations among the most relevant tests for the problem at hand.

Once the neural network is trained and the strategies represented by each node are defined, new performances can be tested on the trained neural network, and the strategy (node) that best matches the new input performance vector can be identified. For instance, if a student were to order many different chemical and physical tests while solving a Hazmat (hazardous materials) case, his performance would be classified with the nodes of the upper right-hand corner of Figure 11, whereas a performance in which the student ordered very few tests would be classified along the left side of the neural network map. Strategies defined in this way can be aggregated by class, grade level, school, or gender and related to other achievement and demographic measures.

3. Hidden Markov Modeling of Problem-Solving Strategy Development

The neural network analysis described in the previous section provided point-in-time snapshots of students' problem-solving strategies and performance. In this section, we describe how longitudinal HMMs were used to model and predict strategic learning trajectories across time and problem sets.

IMMEX problem sets contain a number of isomorphic problems (5–60) for students to solve as they develop different chemistry skill sets. As students performed a series of cases from a problem set, learning trajectories that indicated their progress were developed. First, each student performance in the series was independently classified at the appropriate neural network node as described previously. Second, the sequences of classified student performances became the input to train an HMM. Figure 13 shows the neural network node classifications for four performances of four students. The numbers in the node sequences listed on the right-hand side of the figure correspond directly to the neural network nodes numbered 1–36 in Figure 11.

By mapping these sequences to the performance characteristics at each node of the trained neural network, a profile of each student's progress can be generated. For example, the strategic approaches of students 1 and 3 evolved over time with practice, showing a reduced reliance on background information and progressively refined test selections. Other students showed less strategic adaptation and continued to use the same or similar strategies over time. Manual inspection and mapping of nodes to strategies,

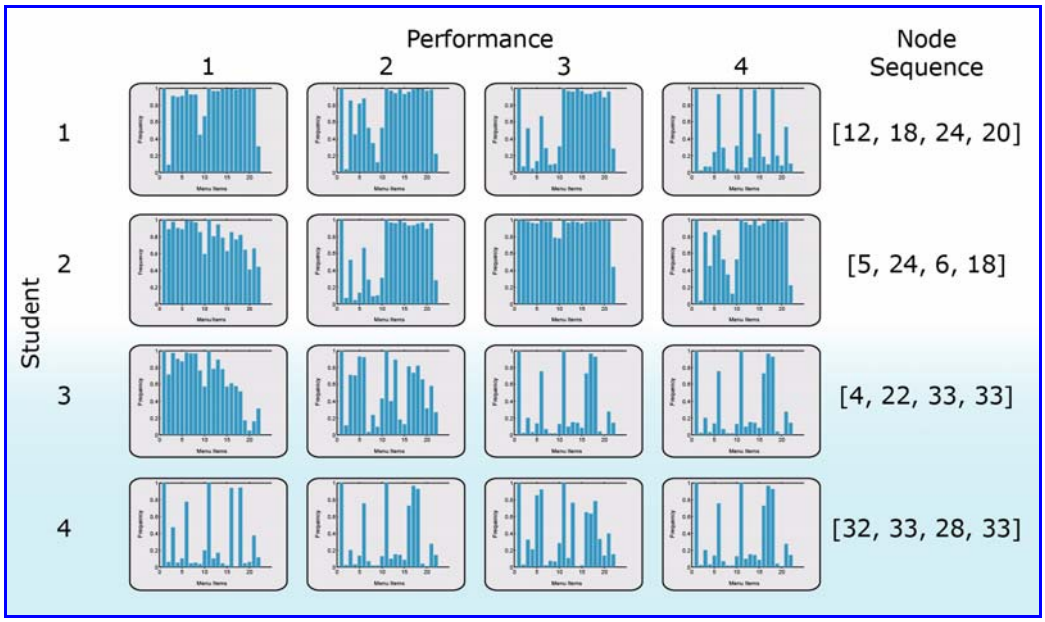

Figure 13. Neural Network Node Classifications for Four Performances of Four Students

while potentially informative, is a time-consuming process. Certainly, Markov models provide an alternative approach for dynamically modeling this longitudinal information; however, the 1,296 possible transitions in a 36-node map render the predictive power of this method less convincing. Instead, HMMs were used to extend our preliminary results to more generally model and predict student learning trajectories.

Figure 14 shows the overall hybrid neural network/HMM methodology. Figure 15 shows an actual trained IMMEX HMM. The HMM training (observation) sequences were given by neural network classifications of different student problem-solving strategies, and the state transitions described the likelihoods of transitioning from one general problem-solving strategy set to another (e.g., on the next problem set). In parallel, this process trained the observation symbol probability distribution, which describes the probabilities of each of the 36 problem-solving strategies (represented by the 36 neural network nodes) at each state. As a student completes a series of IMMEX problem sets, he will typically transit through several HMM states. At each state, the performance is modeled by (a) the general category of problem-solving strategies the student is currently applying (given by the HMM state), (b) his specific strategy (given by the HMM observation, which is linked directly to the 6×6 neural network matrix), and (c) the next most likely strategy the student will apply (given by the HMM state transition matrix).

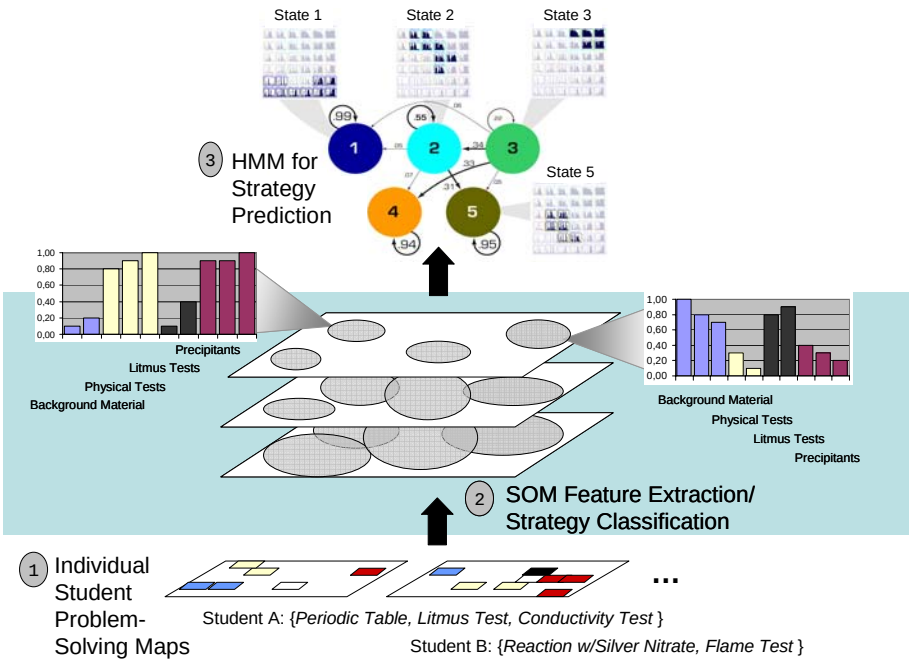

Figure 14. Individual Problem-Solving Maps (Step 1, Bottom) Are Used by SOMs To Identify Students' Problem-Solving Strategies (Step 2, Middle) and Are Then Input to the HMM To Predict Strategy Shifts (Step 3, Top)

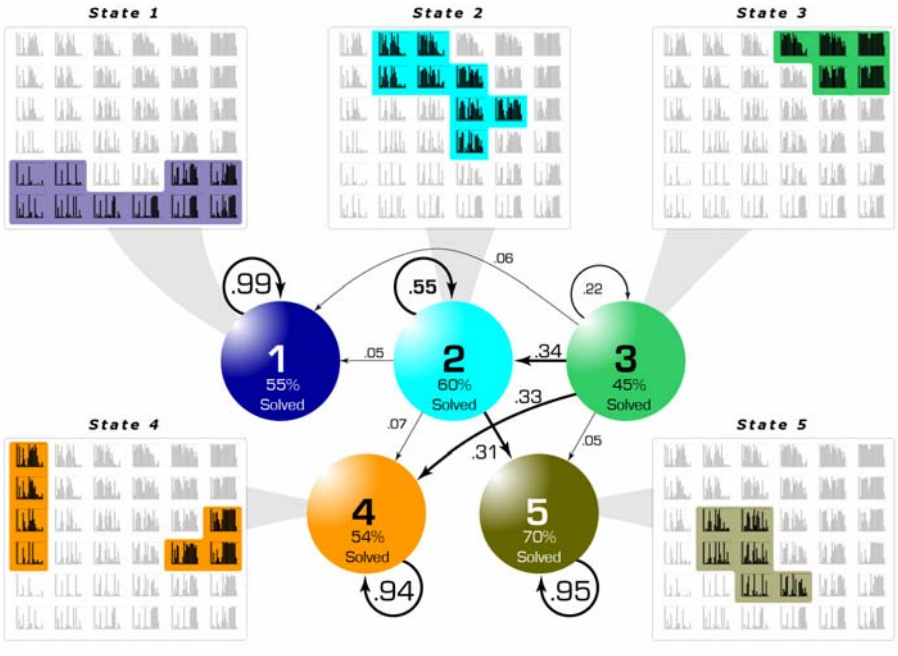

Figure 15. HMM Including State Transition Probabilities and Observation Symbol Probabilities Given by SOMs

Figure 15 illustrates the state transition and observation symbol probabilities obtained from training the HMM with the performances of 1,790 students. The likelihood of transitioning from one state (generalized problem-solving strategy) to another is represented by the probabilities on the labeled arcs in the figure. The state transition probabilities for states 1 (0.99), 4 (0.94), and 5 (0.95) suggest that these states are stable. Once students adopt strategies associated with these states, they are likely to continue to use them. In contrast, students who adopt state 2 and 3 strategies are less likely to persist with those approaches and are more likely to transition to applying other strategies. The highlighted nodes in each map indicate which nodes are most frequently associated with each state. The solution frequencies represent the correct answer on the first attempts.

The trained HMMs thus described patterns of students' strategy shifting over time and could be used to describe and explain learning trajectories and predict future problem-solving performances. For example, we might like to know whether a student is likely to continue using an inefficient problem-solving strategy. This information may enable an instructor to better assess whether the student is likely to need help in the near future.

The overall solution frequency for the testing dataset was 56 percent, and the solution frequencies between the states was significantly different ($\chi^2 = 131.6$, $p < .001$). State 3 had a lower than average solution frequency (45 percent), and State 5 had a higher than average solution frequency (70 percent). The solution frequencies at each state provided an interesting view of progress. For instance, if we compare the differences in solve rates shown with the most likely state transitions from the matrix shown in Figure 15, we see that most of the students who enter state 3 (with the lowest problem-solving rate) will likely transit either to states 2 or 4. Those students who transit from state 3 to state 2 will show on average a 15 percent performance increase (from 45 percent to 60 percent) and those students who transit from state 3 to state 4 will show on average a 9 percent performance increase (from 45 percent to 54 percent). The transition matrix also shows that students who are performing in state 2 (with a 60 percent solve rate) will tend to either stay in that state or transit to state 5, showing a 10 percent performance increase (from 60 percent to 70 percent). This analysis shows that students' performance is increasing and that modeling with neural network and HMM methods enables us to track and understand their learning trajectories.

In a previous section of this paper, we explained how HMMs can be used to predict the likelihood of future behaviors. The prediction accuracy of the IMMEX HMM

was tested by deleting the last known element from the longitudinal sequences of students performances and asking the HMM to predict the likelihood of the missing performance node. For each student performance within a sequence of performances, the most likely corresponding HMM state was calculated. For instance, neural network nodal sequence [6 18 1] mapped to HMM states (3 4 4), meaning that the student started out in state 3, moved to state 4, and then stayed in state 4. Then, the last sequence value was substituted by each of the 36 possible emissions, for instance [18 36 X], where X = 1 to 36. The best predicted value for X was the observation sequence that yielded the maximum path likelihood for the corresponding state sequence, given the HMM. The most likely path probability for each of the 36 possibilities was then compared to the probability of the sequence with the "true" value.

Comparing the "true" values with the predicted values gives an estimate of the predictive power of the model. Figure 16 shows that the prediction power of the HMM increased as student complete more performances. By performances 3, 4, and 5, students were repeatedly using similar strategies, and, by the 6th performance, the model achieved over 90 percent accuracy in predicting the students' next most likely problem-solving strategies.

	HMM Accuracy (% Correct Predictions)
After 1st Performance	67
After 2nd Performance	75
After 3rd Performance	83
After 4th Performance	88
After 5th Performance	86
After 6th Performance	91

Figure 16. HMM Accuracy in Predicting Future Problem-Solving Strategies

The approach described in this section was used to predict students' future behaviors within the IMMEX scientific inquiry environment and provide targeted non-intrusive facilitation. The next section offers recommendations for future work.

E. SUMMARY AND FUTURE DIRECTIONS

This paper presented a basic introduction to some popular stochastic analysis methods from an unbiased, unassociated disciplinary perspective. Examples of these methods were presented through two practical applications of longitudinal stochastic analysis to collaborative and cognitive training environments. The first application, EPSILON, applied a combination of latent mixed Markov modeling and MDS for modeling, analyzing, and supporting the process of online student knowledge sharing. These analysis techniques were used to train a system to dynamically recognize (a) when students are having trouble learning the new concepts they share with each other and (b) why they may be having trouble. In the second application, IMMEX Collaborative, a combination of iterative nonlinear machine learning algorithms was applied to identify latent classes of student problem-solving strategies. The approach was used to predict students' future behaviors within a scientific inquiry environment and to provide targeted nonintrusive facilitation.

When given enough data about a student's previous performances, the IMMEX HMM performed at over 90 percent accuracy when tasked to predict the most likely problem-solving strategy the student will apply next. Knowing whether a student is likely to continue to use an inefficient problem-solving strategy allows us to determine whether the student is likely to need help in the near future. Perhaps more interestingly, however, is the possibility that knowing the distribution of students' problem-solving strategies and their most likely future behaviors may allow us to strategically construct collaborative learning groups containing heterogeneous combinations of various behaviors such that intervention by a human instructor is required less often.

These two applications demonstrated that hybrid combinations of artificial intelligence and statistical mixture models can be used to perform new types of longitudinal analysis of learning and collaboration. Combinations of other types of models, such as neural nets, decision trees, and Bayesian networks have already shown their utility in similar educational assessment applications (e.g., see Mislevy, Steinberg, Breyer, Almond, & Johnson, 1999). Opportunities for exploration along these lines are unlimited. For example, the HMM path probabilities may be used as one factor, among others

obtained statistically, that contributes to a weighted assessment function (Walker, Litman, Kamm, & Abella, 1997) for evaluating student interaction effectiveness. Weighted combinations of factors can also serve as feature vectors in decision trees or input layers in neural networks.

The examples illustrated in this paper are instances of an increasing global trend toward interdisciplinary research. As this trend continues to grow, research that takes advantage of the gaps and overlaps in analytical methodologies between disciplines will save time, effort, and research funds. We should not be surprised to discover that many analytic methods commonly applied within specific disciplines are more widely applicable and adaptable.

REFERENCES

Baum, L. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, *3*, 1–8.

Baum, L., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics, 41,* 164–171.

Bilmes, J. (1998). *A gentle tutorial on the EM algorithm including Gaussian mixtures and Baum-Welch.* ICSI Technical Report TR-97-021 [On-line]. Available: ftp://ftp.icsi.berkeley.edu/pub/techreports/1997/tr-97-021.pdf

Brainerd, C. J. (1979). Markovian interpretations of conservation learning. *Psychological Review, 86,* 181–213.

De Boeck, P., & Wilson, M. (Eds.) (2004). *Explanatory item response models. A generalized linear and nonlinear approach.* New York: Springer.

Fisher, L., & van Belle, G. (1993) *Biostatistics: A methodology for the health sciences*. New York: John Wiley & Sons, Inc.

Goodman, B., Linton, F., Gaimari, R., Hitzeman, J., Ross, H., & Zarrella, G. (2005). Using dialogue features to predict trouble during collaborative learning. *User Modeling and User-Adapted Interaction, 15* (1–2)*,* 85–134.

Greeno, J. (1967). Paired-associate learning with short term retention: Mathematical analysis and data regarding identification of parameters. *Journal of Mathematical Psychology, 4,* 430–472.

Greeno, J., & Steiner, T. (1964). Markovian processes with identifiable states: General considerations and applications to all-or-none learning. *Psychometrika, 29* (4), 309–333.

Hollingshead, A. (1996). The rank-order effect in group decision making. *Organizational Behavior and Human Decision Processes, 68* (3)*,* 181–193.

Juang, B., & Rabiner, L. (1985). A probabilistic distance measure for hidden Markov models. *AT&T Technical Journal, 64* (2), 391–408.

Kintsch, W., & Morris, C. J. (1965). Application of a Markov model for free recall and recognition. *Journal of Experimental Psychology, 69,* 200–206.

Kohonen, T. (2001). *Self-organizing maps* (3rd ed.). New York: Springer.

Kruskal, J., & Wish, M. (1978). *Multidimensional scaling.* Newbury Park, CA: Sage Publications, Inc.

Langeheine, R., & van der Pol, F. (2002). Latent Markov chains. In A. McCutcheon (Ed.), *Advances in latent class models* (pp. 304–341). New York: Cambridge University Press.

Lavery, T., Franz, T., Winquist, J., & Larson, J. (1999). The role of information exchange in predicting group accuracy on a multiple judgment task. *Basic and Applied Social Psychology, 21* (4), 281–289.

Linacre, J. M. (2004). WINSTEPS Rasch measurement computer program [On-line]. Available: http://www.Winsteps.com/

Looney, C. (1997). *Pattern recognition using neural networks: Theory and algorithms for engineers and scientists*. New York: Oxford University Press, Inc.

McCulloch, W., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5,* 115–133.

McCutcheon, A. (1987). *Latent class analysis*. Newbury Park, CA: Sage Publications.

Mennecke, B. (1997). Using group support systems to discover hidden profiles: An examination of the influence of group size and meeting structures on information sharing and decision quality. *International Journal of Human-Computer Studies, 47* (3)*,* 387–405.

Miller, G. (1952). Finite Markov processes in psychology. *Psychometrika, 17* (2)*,* 149–167.

Mislevy, R., Steinberg, L., Breyer, F., Almond, R., & Johnson, L. (1999). A cognitive task analysis, with implications for designing a simulation-based assessment system. *Computers and Human Behavior, 15,* 335–374.

Rabiner, L., (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the Institute of Electrical & Electronics Engineers (IEEE), 77* (2)*,* 257–286.

Ronchetti, M., Gerosa, L., Giordani, A., Soller, A., & Stevens, R. (2005). Symmetric synchronous collaborative navigation applied to e-learning. *IADIS International Journal on WWW/Internet, 3* (1)*,* 1–16.

Schrodt, P. (2000). Pattern recognition of international crises using hidden Markov models. In D. Richards (Ed.), *Political complexity: Nonlinear models of politics* (pp. 296–328). Ann Arbor: University of Michigan Press.

Shepard, R. (1980). Multidimensional scaling, tree-fitting, and clustering. *Science, 210* (4468)*,* 390–398.

Shepard, R., & Arabie, P. (1979). Additive clustering: Representation of similarities as combinations of discrete overlapping properties. *Psychological Review, 86* (2), 87–123.

Smyth, P. (1997). Clustering sequences with hidden Markov models. In M. Mozer, M. Jordan, & T. Petsche (Eds.), *Advances in Neural Information Processing Systems 9* (pp. 648–654). Cambridge, MA: MIT Press.

Soller, A. (2004). Computational modeling and analysis of knowledge sharing in collaborative distance learning. *User Modeling and User-Adapted Interaction, 14* (4), 351–381.

Soller, A., & Lesgold, A. (in press). Modeling the process of knowledge sharing. In U. Hoppe, H. Ogata, and A. Soller (Eds.), *The role of technology in CSCL: Studies in technology enhanced collaborative learning,* (pp. 63–86). Springer.

Soller, A., Martínez-Monés, A., Jermann, P., & Muehlenbrock, M. (2005). From mirroring to guiding: A review of state of the art technology for supporting collaborative learning. *International Journal of Artificial Intelligence in Education, 15* (4)*,* 261–290.

Stasser, G. (1999). The uncertain role of unshared information in collective choice. In L. Thompson, J. Levine, & D. Messick (Eds.), *Shared knowledge in organizations* (pp. 49–69). Hillsdale, NJ: Erlbaum.

Stevens, R., Ikeda, J., Casillas, A., Palacio-Cayetano, J., & S. Clyman (1999). Artificial neural network-based performance assessments. *Computers in Human Behavior, 15* (3)*,* 295–313.

Stevens, R., & Palacio-Cayetano, J. (2003). Design and performance frameworks for constructing problem-solving simulations. *Cell Biology Education, 2* (3)*,* 162–179.

Stevens, R., Soller, A., Cooper, M., & Sprang, M. (2004). Modeling the development of problem-solving skills in chemistry with a Web-based tutor. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems (ITS 2004), Maceió, Brazil,* 580–591.

Stevens, R., Wang, P., & Lopo, A. (1996). Artificial neural networks can distinguish novice and expert strategies during complex problem-solving. *Journal of the American Medical Informatics Association, 3* (2)*,* 131–138.

Vermunt, J. K. (2007). A hierarchical mixture model for clustering three-way data sets. Computational Statistics and Data Analysis, 51 (11), 5368–5376.

Visser, I., Maartje, E., Raijmakers, E. J., & Molenaar, P. (2002). Fitting hidden Markov models to psychological data. *Scientific Programming, 10* (3)*,* 185–199.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory, 13* (2)*,* 260–269.

Walker, M., Litman, D., Kamm, C., & Abella, A. (1997). PARADISE: A framework for evaluating spoken dialogue agents. *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics, Madrid, Spain,* 271–280.

Wickens, T. D. (1982). *Models of behavior: Stochastic processes in psychology.* San Francisco, CA: W. H. Freeman and Company.

Winquist, J. R., & Larson, J. R. (1998). Information pooling: When it impacts group decision making. *Journal of Personality and Social Psychology, 74* (2)*,* 371–377.

Yang, J., Xu, Y., & Chen, C. (1997). Human action learning via hidden Markov model. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, 27* (1), 34–44.

GLOSSARY

3-D	three-dimensional
CRP	Central Research Program
EM	expectation maximization
EPSILON	Encouraging Positive Social Interaction while Learning ON-Line
FFRDC	Federally Funded Research and Development Center
HMM	hidden Markov model
IADIS	International Association for Development of the Information Society
ICSI	International Computer Science Institute
IDA	Institute for Defense Analyses
IEEE	Institute of Electrical & Electronics Engineers
IMMEX	Interactive Multi-Media EXercises
IRT	Item Response Theory
ISODATA	iterative, self-organizing data analysis technique
ITS	Intelligent Tutoring System
KE	knowledge element
LCA	latent class analysis
MDS	multidimensional scaling
MIT	Massachusetts Institute of Technology
OOA&D	Object-Oriented Analysis and Design
SOM	Self-Organizing Map
TR	Technical Report

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED *(From–To)*
April 2007	Final	September 2006 – March 2007

4. TITLE AND SUBTITLE

Applications of Stochastic Analyses for Collaborative Learning and Cognitive Assessment

5a. CONTRACT NUMBER
DASW01-04-C-0003

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Amy Soller
Ron Stevens – UCLA

5d. PROJECT NUMBER

5e. TASK NUMBER
IDA Central Research Project C-2112

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311-1882

8. PERFORMING ORGANIZATION REPORT NUMBER

IDA Document D-3421

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

OSD Studies and FFRDC Programs
4850 Mark Center Drive, Rm. 9604
Alexandria, VA 22311-1882

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited. (27 August 2007)

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This paper presents a basic introduction to some popular stochastic analysis methods from an unbiased disciplinary perspective. Examples ranging from fields as diverse as defense analysis, cognitive science, and instruction, are illustrated throughout this paper to demonstrate the variety of applications that benefit from such stochastic analysis methods and models. We discuss in detail two applications of longitudinal stochastic analysis methods to collaborative and cognitive training environments. The first application applies a combination of Latent Mixed Markov Modeling and Multidimensional Scaling for modeling, analyzing, and supporting the process of online knowledge sharing. In the second application, a combination of iterative nonlinear machine learning algorithms is applied to identify latent classes of problem solving strategies. The examples illustrated in this paper are instances of an increasing global trend toward interdisciplinary re search. As this trend continues to grow, research that takes advantage of the gaps and overlaps in analytical methodologies between disciplines will save time, effort, and research funds.

15. SUBJECT TERMS

assessment, collaboration, educational technology, modeling, probabilistic class analysis, simulation, training

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Robert D. Williams
a. REPORT Uncl.	b. ABSTRACT Uncl.	c. THIS PAGE Uncl.	SAR	45	19b. TELEPHONE NUMBER *(include area code)* 703-845-2192

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18